



파이썬 입력과 출력

학습목표

- 파이썬 자료 입력에 대하여 이해할 수 있습니다.
- 파이썬 `input()` 함수를 이용하여 자료를 입력받고 저장할 수 있습니다.
- 입력받은 자료를 다양한 자료형으로 변환하여 저장할 수 있습니다.
- 파이썬 자료 출력에 대하여 이해할 수 있습니다.
- 파이썬 `print()` 함수의 자료 출력에 대한 다양한 방법을 이해하고 활용할 수 있습니다.

목차

3.1 입력

3.2 출력

3.1 입력

(1) input() 함수를 이용한 표준 입력

예제 3.1 입력의 기본 형태

```
>>> input('이름 입력: ')
이름 입력: 파이썬
'파이썬'
```

(2) input() 함수로 입력 자료 저장

예제 3.2 입력 자료 저장

```
1 name = input('이름 입력: ')
2 print(name)
```

실행결과

```
이름 입력: 파이썬
파이썬
>>>
```

3.1 입력

(3) 문자열로 입력된 자료 (문자열 연결)

예제 3.3 문자열로 입력된 자료

```
1 a = input('첫 번째 숫자 입력: ')
2 b = input('두 번째 숫자 입력: ')
3 print(a+b)
```

실행결과

```
첫 번째 숫자 입력: 10
두 번째 숫자 입력: 20
1020
```

```
>>>
```

3.1 입력

(4) 문자열로 입력 받은 자료를 정수형 변환을 통하여 연산

예제 3.4

정수형으로 입력된 자료

```
1 a = int(input('첫 번째 숫자 입력: '))
2 b = int(input('두 번째 숫자 입력: '))
3 print(a+b)
```

실행결과

첫 번째 숫자 입력: 10

두 번째 숫자 입력: 20

30

>>>

3.1 입력

(5) 문자열로 입력 받은 자료를 실수형 변환을 통한 출력

예제 3.5 실수형으로 입력된 자료

```
1 c = float(input('숫자 입력: '))  
2 print(c)
```

실행결과

숫자 입력: 3.7

3.7

>>>

3.2 출력

(1) print() 함수를 이용한 표준 출력

예제 3.6 출력의 기본 형태

```
>>> print('파이썬으로 출력')
      파이썬으로 출력
>>>
```

(2) 파이썬의 4가지 출력 형태

- 1) 콤마(,)로 구분하여 출력하는 형태
- 2) % 형식지정자를 이용하는 형태
- 3) format() 함수를 이용하는 형태
- 4) f-string 함수를 이용하는 형태

3.2 출력

1) 콤마(,)를 이용한 출력

예제 3.7 콤마(,)로 구분하여 출력

```
1 name = input('이름 입력: ')
2 print('입력하신 이름은', name)
```

실행결과

이름 입력: 파이썬

입력하신 이름은 파이썬

>>>

3.2 출력

2) % 형식지정자를 이용하는 형태

① %s 형식지정자를 이용한 출력

예제 3.8 %s 형식지정자를 이용하여 출력

```
1 name = input('이름 입력: ')
2 print('입력하신 이름은 %s' % name)
```

실행결과

이름 입력: 파이썬

입력하신 이름은 파이썬

>>>

3.2 출력

2) % 형식지정자를 이용하는 형태

② %d 형식지정자를 이용한 출력

예제 3.9 %d 형식지정자를 이용하여 출력

```
1 a = int(input('첫 번째 숫자 입력: '))
2 b = int(input('두 번째 숫자 입력: '))
3 print('%d + %d = %d' % (a, b, a+b))
```

실행결과

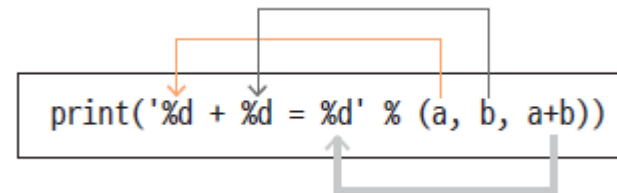
첫 번째 숫자 입력: 10

두 번째 숫자 입력: 20

10 + 20 = 30

>>>

2개 이상의 % 형식지정자를 사용하는 경우에
% 형식지정자와 변수는 순서대로 대응되어 적용



3.2 출력

2) % 형식지정자를 이용하는 형태

③ %f 형식지정자를 이용한 출력

예제 3.10 %f 형식지정자를 이용하여 출력

```
1 a = int(input('첫 번째 숫자 입력: '))
2 b = int(input('두 번째 숫자 입력: '))
3 print('%d + %d = %f' % (a, b, a/b))
```

실행결과

```
첫 번째 숫자 입력: 5
두 번째 숫자 입력: 2
5 / 2 = 2.500000
```

```
>>>
```

%f 를 %.2f 로 수정하면 소수점 이하 2자리까지만 출력하라는 의미

```
print('%d + %d = %.2f' % (a, b, a/b))
```

print() 함수의 % 형식지정자의 종류에는 %s, %d, %f 외에도 여러 가지가 있음

3.2 출력

3) format() 함수를 이용하는 형태

- 출력 내용과 중괄호 { }를 원하는 순서대로 따옴표 안에 작성
- 중괄호에 대응되는 변수는 점(.) 작성 후 format() 함수 안에 순서대로 작성

예제 3.11 format() 함수를 이용하여 출력

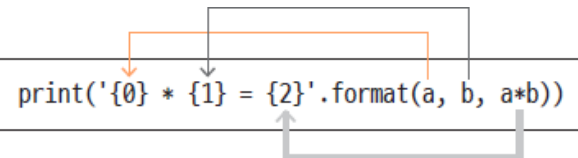
```
1 a = int(input('첫 번째 숫자 입력: '))
2 b = int(input('두 번째 숫자 입력: '))
3 print('{0} x {1} = {2}'.format(a, b, a*b))
```

실행결과

```
첫 번째 숫자 입력: 3
두 번째 숫자 입력: 5
3 x 5 = 15
```

```
>>>
```

- 중괄호와 format() 함수 안의 변수들을 순서대로 대응하여 출력한 형태
- 괄호 안의 번호는 대응되는 변수의 순서로써 인덱스라고 부르며, 0부터 시작함



3.2 출력

3) format() 함수를 이용하는 형태

- 중괄호 순서대로 변수를 대응하여 출력할 때는 인덱스가 생략 가능
- 순서를 변경하여 출력할 때는 인덱스를 원하는 대로 올바르게 작성해야 함
- 자료를 여러 번 반복하여 출력할 때는 반복되기 원하는 자료의 인덱스를 포함한 중괄호를 여러 번 반복하여 작성

예제 3.12 중괄호 순서를 변경하고 자료를 반복하여 출력

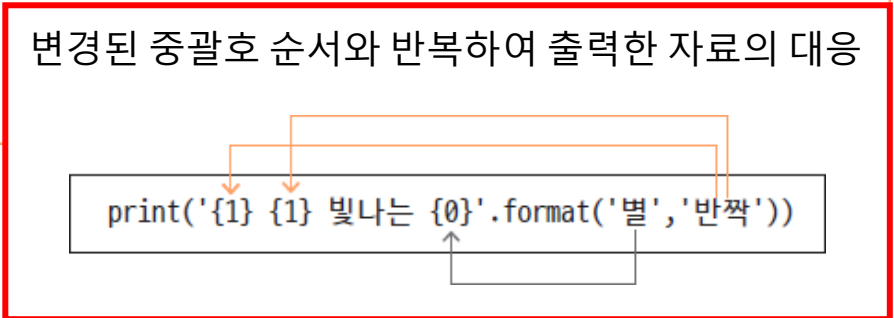
```
1 print('{1} {1} 빛나는 {0}'.format('별', '반짝'))
```

실행결과

반짝 반짝 빛나는 별

>>>

변경된 중괄호 순서와 반복하여 출력한 자료의 대응



```
print('{1} {1} 빛나는 {0}'.format('별', '반짝'))
```

3.2 출력

4) f-string 함수를 이용한 출력

- f-string 포매팅은 파이썬 3.6 버전부터 제공되기 시작
- 문자열 맨 앞에 f를 붙여주고, 중괄호 안에 변수명이나 출력하고 싶은 자료를 직접 작성

예제 3.13 f-string을 이용하여 출력

```
1 menu = '커피'
2 count = 2
3 print(f'{menu}를 좋아해요, 하루 {count}잔 마세요.')
```

실행결과

```
커피를 좋아해요, 하루 2잔 마세요.
>>>
```

TIP

- sep 옵션: print() 함수에서 여러 값을 출력할 때 콤마(,)로 구분되어 있는 값 사이에 기본적으로 공백(스페이스)이 들어갑니다. 하지만 sep 옵션을 사용하면 공백 대신 다른 문자를 지정할 수 있습니다.

예 print('Hello', 'World', sep=',')

결과 HelloWorld

- end 옵션: print() 함수는 기본적으로 출력 후 줄바꿈을 추가합니다. 이를 변경하고 싶다면 end 옵션을 사용하여 줄바꿈 대신 원하는 문자열을 지정할 수 있습니다.

예 print('Hello', end=' ')

결과 Hello (줄바꿈 없이 공백이 추가됨)

수고하셨습니다.

